

## REMARKS

In the Office Action, the Examiner rejected the claims under 35 USC §101, 35 USC §112 first and second paragraphs, and under 35 USC §103. In addition, the Examiner objected to the title. These objections and rejections are fully traversed below.

The title has been amended corresponding to the Examiner's recommendations. In addition, the claims have been amended to correct minor informalities and to further clarify the subject matter regarded as the invention. Claims 1-25 and 27-29 remain pending.

Reconsideration of the application is respectfully requested based on the following remarks.

### REJECTION OF CLAIMS UNDER 35 USC §101

The claims have been rejected under 35 U.S.C. §101 because the claimed invention is allegedly directed to non-statutory subject matter. Applicant respectfully traverses this assertion. The claimed invention relates to a computer-related process that has a practical application in the technological arts. Specifically, the computer-related process relates to execution of a set of source code instructions. Specifically, the computer-related process identifies sections of source code that generate either local objects or non-local objects when the source code is executed, and stores these local/non-local objects in separate heaps of memory. As such, Applicant respectfully submits that claims are directed to statutory subject matter, and respectfully requests that the Examiner withdraw the rejection to the claims under 35 U.S.C. § 101.

### REJECTION OF CLAIMS UNDER 35 USC §112

In the Office Action, the Examiner rejected the claims under 35 USC §112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the

subject matter which Applicant regards as the invention. In view of the currently amended claims, Applicant respectfully submits that the claims distinctly claim the subject matter regarded as the invention.

The Examiner indicates that “the data structure” is indefinite. Applicant respectfully submits that “a data structure” as initially recited in the first element of claim 1 is not indefinite. The phrase “a data structure” is recited in the first element of each independent claim (e.g., claim 1). Thus, there is only one data structure recited in the claims, which is used to store addresses associated with source code instructions that generate local objects, when executed. The next source code instruction is obtained from the set of source code instructions, as claimed, not from a data structure. Source code is generally stored in a file from which successive lines (e.g., “next source code instruction”) are executed. Such a file need not be explicitly claimed in order to particularly point out and distinctly claim the use of the set of source code instructions. Moreover, these successive lines need not be separately stored in order to execute (or interpret) them.

The method claimed in claim 1 is one that executes a method, which is a term well-known in the art of software programming. However, in order to clarify the subject matter being claimed, Applicant has amended this term to the phrase “a set of source code instructions.” Method execution, as claimed, is described in detail in FIG. 5. Thus, FIG. 5 and associated text details one example of executing the method using information stored in a data structure such as that generated in FIG. 2 or FIG. 4. Specifically, the method is executed such that local objects and non-local objects are stored in separate heaps. While this invention is illustrated with reference to a Java frame, this example is merely illustrative. Specifically, as recited in the claims, the intermediate code generated during compile time may be referred to as source code (intermediate source code) or bytecodes. During execution, the next source code (bytecode) of the source code is obtained and interpreted (in this example). During the interpretation, the objects are stored according to whether they are classified in the data structure (e.g., storing a PC or address of the bytecodes that generate the local objects during execution of the method) as local or non-local objects. Accordingly, in view of the above, Applicant respectfully requests that the Examiner withdraw the rejection to the claims under 35 USC §112, second paragraph.

In addition, the Examiner rejected the claims under 35 USC §112, first paragraph. Applicant respectfully asserts that, in view of the claim amendments amending the term “method” to “a set of source code instructions,” the summary on page 4 of the specification as well as the detailed description support the claimed invention. FIG. 1 and associated text illustrate the process during which the source code (e.g., intermediate source code or “bytecodes”) and data structure identifying addresses of the portions of the source code that create local objects are generated. Specifically, intermediate source code (source code) is generated from the method (e.g., initial source code) at block 104. Generation of the data structure (optimization) is performed at block 106 (and further detailed in FIG. 2). The data structure (e.g., local table) is generated at block 204 of FIG. 2 and further illustrated in FIG. 3. FIG. 4 details one specific way to generate class files (and associated attribute\_info structure of a Java class file). In other words, the attribute\_info structure is one manner of implementing the data structure claimed (e.g., of FIG. 2).

### **REJECTION OF CLAIMS UNDER 35 USC §103**

In the Office Action, the Examiner rejected the claims under 35 USC §103 as being unpatentable over Pradhan et al., U.S. Patent No. 6,446,257, (‘Pradhan’ hereinafter).

The present invention enables local and non-local objects to be stored in separate heaps, thereby optimizing the garbage collection process. Pradhan neither discloses nor suggests the claimed invention. Pradhan does relate to garbage collection. However, Pradhan says nothing about local objects. Rather, col. 2, lines 59-60 of Pradhan indicate that the invention relates to “live object relocation.” While a live object is referred to as an object that “may be globally known” in col. 1, lines 34-35, there is no reference to local and non-local objects, or the separate storage of local and non-local objects. While the Examiner cites col. 8, lines 2-4 (source code that creates local objects) and col. 9, lines 26-30 (creating a local object on a local heap of memory), Applicant respectfully submits that col. 8, lines 2-4 merely state “A source compiler 107 processes a source code file 118 and thereby transforms the source code file 118 into an intermediate file 122.” Moreover, col. 9, lines 26-30 merely state that “garbage collection works in conjunction with heap allocation...” In no manner does Pradhan disclose or suggest storing local and non-local objects in separate heaps. In fact, Pradhan discloses that a heap may be divided into two or more groups segregated by age, where the groups are referred to as generations. See Pradhan, col. 9, lines 36-40. Since

segregation by age is not identical to segregation of local and non-local objects, Applicant respectfully submits that the claimed invention is not anticipated by Pradhan.

The Examiner admits that Pradhan does not expressly teach storing two heaps for local and non-local objects. The presently pending claims enable the memory associated with the local heap to be released or reclaimed upon termination of execution of the method. For instance, claim 3 recites “reclaiming memory associated with the local heap upon termination of execution of the set of source code instructions.” Similarly, claim 4 recites “returning memory associated with the local heap to a pool of available memory upon termination of execution of the set of source code instructions.” This is accomplished through storing local objects on a local heap of memory and non-local objects on a non-local heap of memory.

The Examiner has taken official notice that storing local and non-local objects in separate data structures or memories was common and notoriously well known to the ordinary skilled in the computer art. The Examiner further asserts that it is inconvenient to store local and non-local variables in the same data structure because the system is less efficient, and states that one would have stored local and non-local objects in two distinct heaps to increase input/output processing. Applicant respectfully traverses these assertions and the Examiner’s taking of official notice. Specifically, Applicant cites the Background section of Applicant’s specification, which states that “these garbage collection optimization schemes treat all objects equally, rather than treating each type of object differently.” See page 2, lines 16-17. Applicant therefore respectfully traverses the Examiner’s taking of official notice that storing local and non-local objects in separate data structures or memories was common and notoriously well known to the ordinary skilled in the computer art. In addition, the Examiner further argues that “Pradhan clearly suggests using two heaps for local and non-local objects.” Applicant respectfully traverses these assertions. Rather, Applicant asserts that the Examiner is applying hindsight in these assertions, since Pradhan merely discloses segregating groups by age and neither discloses nor suggests storing local and non-local objects in separate heaps. Accordingly, Applicant respectfully submits that the claims are patentable over the cited art.

The dependent claims depend from one of the independent claims and are therefore patentable for at least the same reasons. However, the dependent claims recite additional limitations that further distinguish them from the cited references. For instance, dependent

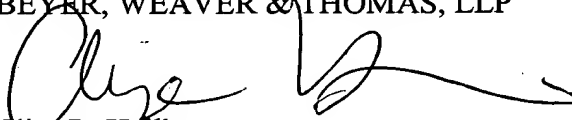
claim 13 recites that the data structure is an attribute info structure of a Java class file. Hence, it is submitted that the dependent claims are patentably distinct from Pradhan.

Based on the foregoing, it is submitted that the independent claims are patentably distinct from Pradhan. In addition, it is submitted that the dependent claims are also patentably distinct for at least the same reasons. The additional limitations recited in the independent claims or the dependent claims are not further discussed as the above discussed limitations are clearly sufficient to distinguish the claimed invention from Pradhan. Thus, it is respectfully requested that the Examiner withdraw the rejection of the claims under 35 USC §103.

Reconsideration of the application and an early Notice of Allowance are earnestly solicited. If there are any issues remaining which the Examiner believes could be resolved through either a Supplemental Response or an Examiner's Amendment, the Examiner is respectfully requested to contact the undersigned attorney at the telephone number listed below.

Applicants hereby petition for an extension of time which may be required to maintain the pendency of this case, and any required fee for such extension or any further fee required in connection with the filing of this Amendment is to be charged to Deposit Account No. 50-0388 (Order No. SUN1P287).

Respectfully submitted,  
BEYER, WEAVER & THOMAS, LLP



Elise R. Heilbrunn  
Reg. No. 42,649

BEYER, WEAVER & THOMAS, LLP  
P.O. Box 778  
Berkeley, CA 94704-0778  
Tel. (510) 843-6200